**学习项目**

|  |
| --- |
| “Java 编程语言基础”实验报告  [基础语法（上）](https://www.shiyanlou.com/courses/1230/) 关系运算符 关系运算符生成的是一个 boolean（布尔）结果，它们计算的是操作数的值之间的关系。如果关系是真实的，结果为 true（真），否则，结果为 false（假）。  表格中的例子中，假设变量 a 为 3，变量 b 为 5：  图片描述  除了上表列出的二元运算符，Java 还有唯一的一个三目运算符 ?: 。  语法格式： 布尔表达式？表达式 1 : 表达式 2  运算过程：如果布尔表达式的值为true ，则返回 表达式1 的值，否则返回 表达式2 的值。 运算符优先级 运算符的优先级是帮助我们在一个表达式中如何对于不同的运算符和相同的运算符，进行正确的运算顺序。  运算符的优先级不需要特别地去记忆它，比较复杂的表达式一般使用圆括号()分开，提高可读性。 练习：计算数字和 import java.util.Scanner;  public class Sum {  public static void main(String args[]) {  Scanner in =new Scanner(System.in);  //获取int值  int x1=in.nextInt();  int x2=in.nextInt();  System.out.println(x1+x2);  in.close();  }  } 关键字和语句 Java 的关键字对 java 的编译器有特殊的意义，他们用来表示一种数据类型，或者表示程序的结构等，关键字不能用作变量名、方法名、类名、包名。  Java 关键字有如下表所列，目前共有 50 个 Java 关键字，其中，"const"和"goto"这两个关键字在 Java 语言中并没有具体含义。  图片描述 方法 方法的定义语法：  访问修饰符 返回值类型 方法名(参数列表){  方法体  }  1.访问修饰符：代表方法允许被访问的权限范围， 可以是 public、protected、private 或者省略（default） ，其中 public 表示该方法可以被其他任何代码调用。  2.返回值类型：方法返回值的类型，如果方法不返回任何值，则返回值类型指定为 void (代表无类型)；如果方法具有返回值，则需要指定返回值的类型，并且在方法体中使用 return 语句返回值。  3.方法名：是方法的名字，必须使用合法的标识符。  4.参数列表：是传递给方法的参数列表，参数可以有多个，多个参数间以逗号隔开，每个参数由参数类型和参数名组成，以空格隔开。当方法被调用时，传递值给参数。这个值被称为实参或变量。参数列表是指方法的参数类型、顺序和参数的个数。参数是可选的，方法可以不包含任何参数。  5.方法体：方法体包含具体的语句，定义该方法的功能。  根据方法是否带参、是否带返回值，可将方法分为四类：  无参无返回值方法 无参带返回值方法 带参无返回值方法 带参带返回值方法 当方法定义好之后，需要调用才可以生效，我们可以通过 main 方法（mian 方法是 Java 程序的入口，所以需要用它来调用）来调用它 练习题：方法使用 在/home/project/目录下新建文件MethodTest.java，在其中新建一个方法methodDemo，运行该方法，在控制台输出Hello Shiyanlou。  public class MethodTest{  public static void main(String[] args){  MethodTest.methodDemo();  }  public static void methodDemo(){  System.out.println("Hello Shiyanlou");  }  } |

|  |
| --- |
| “Linux 基础入门（新版）”实验报告 [基本概念及操作](https://www.shiyanlou.com/courses/1/) Linux 系统简介 终端本质上是对应着 Linux 上的 /dev/tty 设备，Linux 的多用户登陆就是通过不同的 /dev/tty 设备完成的，Linux 默认提供了 6 个纯命令行界面的 “terminal”（准确的说这里应该是 6 个 virtual consoles）来让用户登录。在物理机系统上你可以通过使用[Ctrl]+[Alt]+[F1]～[F6]进行切换，不过在我们的在线实验环境中可能无法切换，因为特殊功能按键会被你的主机系统劫持。当你切换到其中一个终端后想要切换回图形界面，你可以按下[Ctrl]+[Alt]+[F7]来完成。 Shell UNIX/Linux 操作系统下的 Shell 既是用户交互的界面，也是控制系统的脚本语言。当然这一点也有别于 Windows 下的命令行，虽然该命令行也提供了很简单的控制语句。在 Windows 操作系统下，有些用户从来都不会直接使用 Shell，然而在 UNIX 系列操作系统下，Shell 仍然是控制系统启动、X11 启动和很多其它实用工具的脚本解释程序。  在 UNIX/Linux 中比较流行的常见的 Shell 有 bash、zsh、ksh、csh 等等，Ubuntu 终端默认使用的是 bash，默认的桌面环境是 GNOME 或者 Unity（基于 GNOME），但我们的环境中使用的分别是 zsh 和 xfce。 命令行操作体验 在linux中，最最重要的就是命令，这就包含了2个过程，输入和输出  输入：输入当然就是打开终端，然后按键盘输入，然后按回车，输入格式一般就是这类的 创建一个名为 file 的文件，touch是一个命令 $ touch file  进入一个目录，cd是一个命令 $ cd /etc/  查看当前所在目录 $ pwd 输出：输出会返回你想要的结果，比如你要看什么文件，就会返回文件的内容。如果只是执行，执行失败会告诉你哪里错了，如果执行成功那么会没有输出，因为linux的哲学就是：没有结果就是最好的结果  1) 重要快捷键：  真正学习命令行之前，你先要掌握几个十分有用、必需掌握的小技巧：  [Tab]  使用Tab键来进行命令补全，Tab键一般是在字母Q旁边，这个技巧给你带来的最大的好处就是当你忘记某个命令的全称时可以只输入它的开头的一部分，然后按下Tab键就可以得到提示或者帮助完成  当然不止补全命令，补全目录、补全命令参数都是没问题的  [Ctrl+c]  想想你有没有遇到过这种情况，当你在 Linux 命令行中无意输入了一个不知道的命令，或者错误地使用了一个命令，导致在终端里出现了你无法预料的情况，比如，屏幕上只有光标在闪烁却无法继续输入命令，或者不停地输出一大堆你不想要的结果。你想要立即停止并恢复到你可控的状态，那该怎么办呢？这时候你就可以使用Ctrl+c键来强行终止当前程序（你可以放心它并不会使终端退出）。  其他一些常用快捷键  按键 作用 Ctrl+d 键盘输入结束或退出终端 Ctrl+s 暂停当前程序，暂停后按下任意键恢复运行 Ctrl+z 将当前程序放到后台运行，恢复到前台为命令fg Ctrl+a 将光标移至输入行头，相当于Home键 Ctrl+e 将光标移至输入行末，相当于End键 Ctrl+k 删除从光标所在位置到行末 Alt+Backspace 向前删除一个单词 Shift+PgUp 将终端显示向上滚动 Shift+PgDn 将终端显示向下滚动  2) 学会利用历史输入命令  很简单，你可以使用键盘上的方向上键↑，恢复你之前输入过的命令，你一试便知。  3) 学会使用通配符  通配符是一种特殊语句，主要有星号（\*）和问号（?），用来对字符串进行模糊匹配（比如文件名、参数名）。当查找文件夹时，可以使用它来代替一个或多个真正字符；当不知道真正字符或者懒得输入完整名字时，常常使用通配符代替一个或多个真正字符。  终端里面输入的通配符是由 Shell 处理的，不是由所涉及的命令语句处理的，它只会出现在命令的“参数值”里（它不能出现在命令名称里， 命令不记得，那就用Tab补全）。当 Shell 在“参数值”中遇到了通配符时，Shell 会将其当作路径或文件名在磁盘上搜寻可能的匹配：若符合要求的匹配存在，则进行代换（路径扩展）；否则就将该通配符作为一个普通字符传递给“命令”，然后再由命令进行处理。总之，通配符实际上就是一种 Shell 实现的路径扩展功能。在通配符被处理后， Shell 会先完成该命令的重组，然后继续处理重组后的命令，直至执行该命令。  图片描述  在创建文件的时候，如果需要一次性创建多个文件，比如：“love\_1\_linux.txt，love\_2\_linux.txt，... love\_10\_linux.txt”。在 Linux 中十分方便：  $ touch love\_{1..10}\_shiyanlou.txt  Shell 常用通配符：  字符 含义   * 匹配 0 或多个字符 ? 匹配任意一个字符 [list] 匹配 list 中的任意单一字符 [^list] 匹配 除list 中的任意单一字符以外的字符 [c1-c2] 匹配 c1-c2 中的任意单一字符 如：[0-9] [a-z] {string1,string2,...} 匹配 string1 或 string2 (或更多)其一字符串 {c1..c2} 匹配 c1-c2 中全部字符 如{1..10}   图片描述  图片描述  4) 学会在命令行中获取帮助  在 Linux 环境中，如果你遇到困难，可以使用man命令，它是Manual pages的缩写。  Manual pages 是 UNIX 或类 UNIX 操作系统中在线软件文档的一种普遍的形式， 内容包括计算机程序（包括库和系统调用）、正式的标准和惯例，甚至是抽象的概念。用户可以通过执行man命令调用手册页。  你可以使用如下方式来获得某个命令的说明和使用方式的详细介绍：  $ man <command\_name>  比如你想查看 man 命令本身的使用方式，你可以输入：  man man  通常情况下，man 手册里面的内容都是英文的，这就要求你有一定的英文基础。man 手册的内容很多，涉及了 Linux 使用过程中的方方面面。为了便于查找，man 手册被进行了分册（分区段）处理，在 Research UNIX、BSD、OS X 和 Linux 中，手册通常被分为8个区段，安排如下：  区段 说明 1 一般命令 2 系统调用 3 库函数，涵盖了C标准函数库 4 特殊文件（通常是/dev中的设备）和驱动程序 5 文件格式和约定 6 游戏和屏保 7 杂项 8 系统管理命令和守护进程 要查看相应区段的内容，就在 man 后面加上相应区段的数字即可，如：  $ man 1 ls  图片描述  会显示第一区段中的ls命令 man 页面。  所有的手册页遵循一个常见的布局，为了通过简单的 ASCII 文本展示而被优化，而这种情况下可能没有任何形式的高亮或字体控制。一般包括以下部分内容：  NAME（名称）  该命令或函数的名称，接着是一行简介。 SYNOPSIS（概要）  对于命令，正式的描述它如何运行，以及需要什么样的命令行参数。对于函数，介绍函数所需的参数，以及哪个头文件包含该函数的定义。 DESCRIPTION（说明）  命令或函数功能的文本描述。 EXAMPLES（示例）  常用的一些示例。 SEE ALSO（参见）  相关命令或函数的列表。 也可能存在其它部分内容，但这些部分没有得到跨手册页的标准化。常见的例子包括：OPTIONS（选项），EXIT STATUS（退出状态），ENVIRONMENT（环境），BUGS（程序漏洞），FILES（文件），AUTHOR（作者），REPORTING BUGS（已知漏洞），HISTORY（历史）和 COPYRIGHT（版权）。  通常 man 手册中的内容很多，你可能不太容易找到你想要的结果，不过幸运的是你可以在 man 中使用搜索/<你要搜索的关键字>，查找完毕后你可以使用n键切换到下一个关键字所在处图片描述  shift+n为上一个关键字所在处。使用Space（空格键）翻页，Enter（回车键）向下滚动一行，或者使用k,j（vim 编辑器的移动键）进行向前向后滚动一行。按下h键为显示使用帮助（因为 man 使用 less 作为阅读器，实为less工具的帮助），按下q退出。  图片描述  想要获得更详细的帮助，你还可以使用info命令，不过通常使用man就足够了。如果你知道某个命令的作用，只是想快速查看一些它的某个具体参数的作用，那么你可以使用--help参数，大部分命令都会带有这个参数，如：  ls --help 作业 一个有趣的 Linux 命令，这一节介绍一个可以输出图形字符的命令banner  banner jx8zjs  使用前先安装、  $ sudo apt-get update  $ sudo apt-get install sysvbanner  图片描述  还可以使用默认已经安装的一个命令printerbanner：  $ printerbanner -w 60 B  插入代码片段  图片描述  -w参数指定打印宽度，因为我们的环境在屏幕中显示比较小，必须要加上宽度限制。  还有两个类似的命令toilet,figlet，作为作业安装试用。  sudo apt-get install toilet  sudo apt-get install figlet  图片描述  图片描述 |

学习心得

Linux快捷键确实很好用，但是功能很多也需要大量的实践才能真正提升使用效率。

**今日完成**

Java 编程语言基础基础语法（上）

Linux 系统简介和基础操作

**待解决问题**

暂无

**其他**

暂无

**明日计划：**

最近准备答辩后事宜。

明天完成基础语法下，继续linux学习